

**Progettazione di Applicazioni Web T**  
**Prova d'Esame – 14 luglio 2022**

**Tempo a disposizione: 2 ore e 30 minuti**

---

La soluzione comprende la consegna elettronica dei seguenti file:

<b>Hotel.zip</b>	file zip contenente il sorgente java/class, file XML/XSD e txt per il punto 1
<b>Archivio.zip</b>	file zip contenente il sorgente java/class/XML e file txt per il punto 2
<b>WebRatio.zip</b>	file zip contenente il testo di risposta (txt/doc/pdf) al punto 3

**Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, codice java e relativi .class, descrittori XML, file txt/doc/pdf, ecc.) e NON dell'intero progetto.**

**N.B. La prova si intende superata se il punteggio finale non è inferiore a 18 punti (su un totale di 33). In particolare, è necessario (ma non sufficiente) ottenere la sufficienza in tutti gli esercizi.**

---

**ESERCIZIO 1 - 14 punti (sufficienza: 7 punti)**

Si progetti una **grammatica XML Schema**, assieme a un suo **documento XML** di esempio, in grado di modellare le informazioni relative a un **sito Web per la ricerca di Alberghi** nel rispetto delle seguenti specifiche:

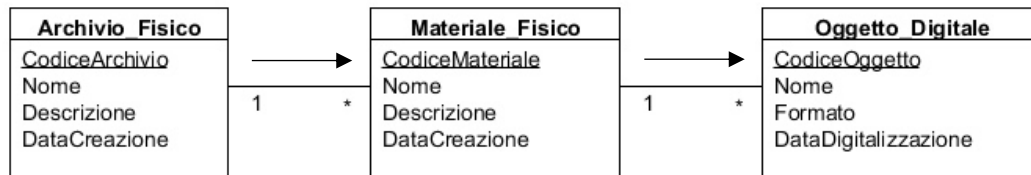
- Ciascun documento XML modella una ricerca di un utente registrato al sito Web.
- Ogni ricerca tiene traccia di informazioni utente quali e-mail (di tipo “email” e obbligatorio) e password (di tipo “password” e obbligatorio) e di attributi per specificare una ricerca da parte dell’utente (in particolare, “nazione”, “città”, “numero stelle” e “fascia prezzo” - tutti obbligatori, tranne fascia prezzo che è opzionale).
- Nel dettaglio, “nazione” e “città” sono stringhe, “numero stelle” è un intero compreso tra 1 e 5, mentre “fascia prezzo” rappresenta il costo/notte dell’albergo in Euro e può appartenere a 4 possibili fasce: bassa (inferiore a €50), media (tra €50 e €120), alta (da €121 a €180) e lusso superiore a €180).
- Il tipo “password” è una stringa di 8 caratteri che deve contenere almeno un numero e un carattere speciale tra {?!\*}\$}.
- Il tipo “email” è una stringa formata da due sottostringhe separate dal carattere speciale “@” e in cui la seconda sottostringa contiene un “.”.
- Il risultato della ricerca è modellato a sua volta come un insieme di “oggetti” complessi (cardinalità dell’insieme da 0 a N), ognuno formato da “nome hotel” (tipo stringa), “fotografia hotel” (rappresentata dal nome del file jpeg corrispondente), “giudizio utenti”, tra {eccellente, buono, medio, sufficiente, mediocre}, “prezzo” (espresso in Euro) e “link al sito Web hotel dedicato” (tipo indirizzo Web). Tutti i campi sono obbligatori. Durante la navigazione del risultato, l’utente può visitare il link del sito Web dedicato dell’albergo di interesse.

Si realizzi quindi l’applicazione Java “IndagineMercatoHotel” che, facendo uso del **parser SAX** e del **documento XML di esempio** realizzato al punto precedente, esponga il metodo `getSelezione()`, unitamente a un suo `main` di prova, in grado di restituire i nomi degli alberghi (associati al risultato della ricerca) per i quali l’utente ha fatto “click” sul link al sito Web dedicato. Si stampi tale lista sul file **Hotel.txt**.

**Progettazione di Applicazioni Web T**  
**Prova d'Esame – 14 luglio 2022**

**ESERCIZIO 2 - 14 punti (sufficienza: 7 punti)**

Partendo dalla realtà illustrata nel **diagramma UML** di seguito riportato, si fornisca una soluzione alla gestione della persistenza basata su **metodologia Hibernate** in grado di “mappare” efficientemente, e con uso di ID surrogati, il modello di dominio rappresentato dai **JavaBean Archivio\_Fisico, Materiale\_Fisico e Oggetto\_Digitale** del **diagramma UML** con le corrispondenti **tabelle relazionali derivate dalla progettazione logica** del diagramma stesso.



Nel dettaglio, dopo aver creato da applicazione Java gli **schemi delle tabelle** all'interno del proprio schema nel database **TW\_STUD** di **DB2** (esplicitando tutti i vincoli opportuni), implementato i **JavaBeans**, definiti i **file XML di mapping** e il **file XML di properties**, si richiede la realizzazione di una **classe di prova** facente uso delle **API Hibernate** in grado di:

- inserire due o più tuple nelle tabelle di interesse;
- restituire:
  - per ogni archivio fisico post 2014, il numero dei formati distinti in cui sono raggruppati i corrispondenti oggetti digitalizzati;
  - nomi degli archivi con formati digitali di tipo “jpeg”;
- produrre una stampa opportunamente formattata e completa dei risultati delle query del punto precedente sul file **Archivio.txt**;

il tutto, mediante opportuna **gestione delle transazioni**. Il candidato è invitato a motivare opportunamente le proprie scelte in fase di demarcazione delle transazioni mediante commento da riportare sullo stesso file **Archivio.txt**.

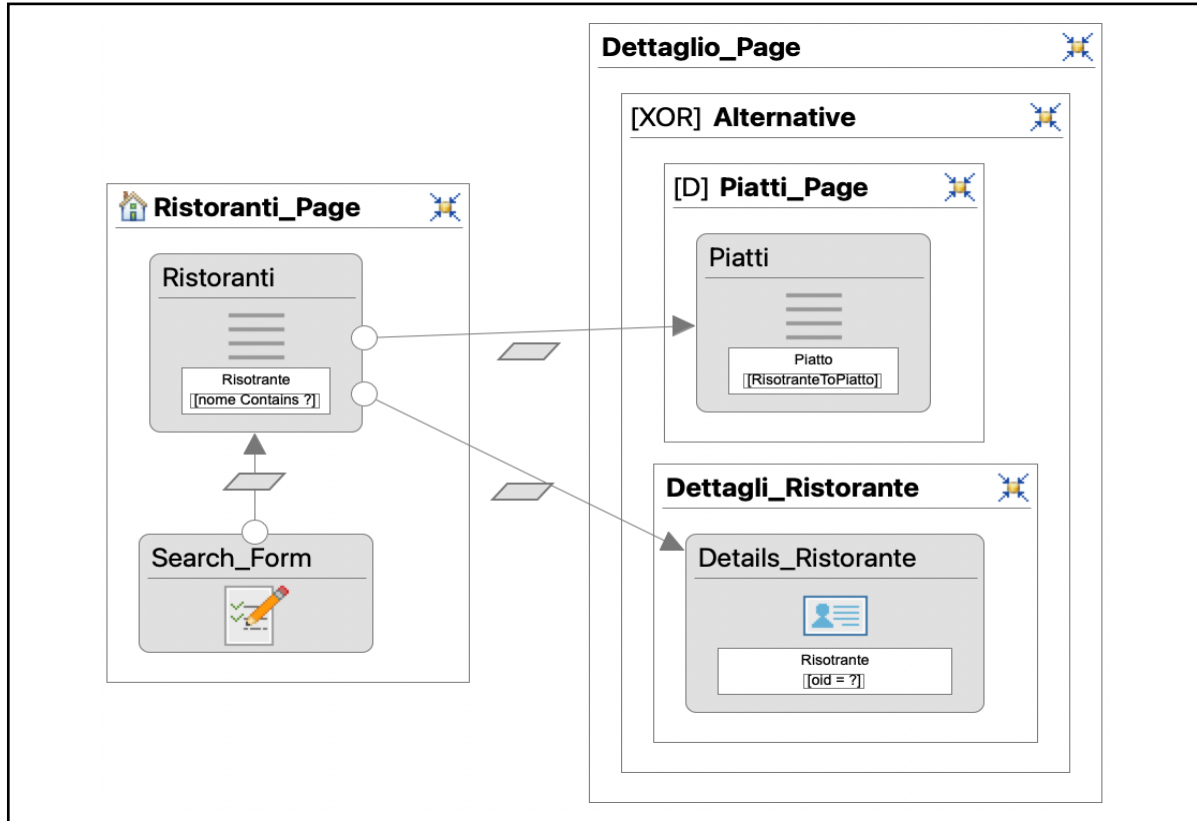
**N.B.** L'implementazione **deve limitarsi** al solo **DBMS DB2**. La soluzione **deve sfruttare i mapping specificati nello schema UML**. Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.

**Progettazione di Applicazioni Web T**  
**Prova d'Esame – 14 luglio 2022**

**ESERCIZIO 3 - 5 punti (sufficienza: 3 punti)**

Considerando l'ipertesto progettato nello **schema IFML di WebRatio** di seguito riportato,

**SV**



il candidato descrive dettagliatamente le relazioni esistenti tra le componenti pagina, link e condizioni espresse sulle componenti della *site view SV*.